

# Message passing with relaxed moment matching \*

Yuan Qi

Departments of CS and Statistics

Purdue University

West Lafayette, IN 47907

[alanqi@cs.purdue.edu](mailto:alanqi@cs.purdue.edu)

Yandong Guo

School of ECE

Purdue University

West Lafayette, IN 47907

[guoy@purdue.edu](mailto:guoy@purdue.edu)

March 2, 2013

## Abstract

Bayesian learning is often hampered by large computational expense. As a powerful generalization of popular belief propagation, expectation propagation (EP) efficiently approximates the exact Bayesian computation. Nevertheless, EP can be sensitive to outliers and suffer from divergence for difficult cases. To address this issue, we propose a new approximate inference approach, relaxed expectation propagation (REP). It relaxes the moment matching requirement of expectation propagation by adding a relaxation factor into the KL minimization. We penalize this relaxation with a  $l_1$  penalty. With this penalty, when two distributions in the relaxed KL divergence are similar, we obtain the exact moment matching; in the presence of outliers, the relaxation factor will be used to relax the moment matching constraint. Based on this penalized KL minimization, REP is robust to outliers and can greatly improve the posterior approximation quality over EP. To examine the effectiveness of REP, we apply it to Gaussian process classification, a task known to be suitable to EP. Our classification results on synthetic and UCI benchmark datasets demonstrate significant improvement of REP over EP and Power EP—in terms of algorithmic stability, estimation accuracy and predictive performance.

**Keywords:** Approximate Bayesian inference, Relaxed moment matching, Expectation propagation,  $l_1$  penalty, Gaussian process classification

## 1 Introduction

Bayesian learning provides a principled framework for modeling complex systems and making predictions. A critical component of Bayesian learning is the computation of posterior

---

\*This work was sponsored by the NSF grants IIS-0916443, IIS-1054903, ECCS-0941533, and CCF-0939370. All the authors gratefully acknowledge the support of the grants. Any opinions, findings, and conclusion or recommendation expressed in this material are those of the author(s) and do not necessarily reflect the view of the funding agencies or the U.S. government.

distributions that represent estimation uncertainty. However, the exact computation is often so expensive that it has become a bottleneck for practical applications of Bayesian learning. To address this challenge, a variety of approximate inference methods has been developed to speed up the computation [Jaakkola, 2000, Minka, 2001, Opper and Winther, 2005, Wainwright and Jordan, 2008]. As a representative approximate inference method, expectation propagation [Minka, 2001] generalizes the popular belief propagation algorithm, allows us to use structured approximations and handles both discrete and continuous posterior distributions. EP has been shown to significantly reduce computational cost while maintaining high approximation accuracy; for example, Kuss and Rasmussen [2005] have demonstrated that, for Gaussian process (GP) classification, EP can provide accurate approximation to predictive posteriors.

Despite its success in many applications, EP can be sensitive to outliers in observation and suffer from divergence when the exact distribution is not close to the approximating family used by EP. This stems from the fact that EP approximates each factor in the model by a simpler form, known as messages, and iteratively refines the messages (See Section 2). Each message refinement is based on moment matching, which minimizes the Kullback-Leibler (KL) divergence between old and new beliefs. The messages are refined in a distributed fashion—resulting in efficient inference on a graphical model. But when the approximating family cannot fit the exact posterior well—such as in the presence of outliers—the message passing algorithm can suffer from divergence and give poor approximation quality.

We can force EP to converge by using the CCCP algorithm [Yuille, 2002, Heskes et al., 2005]. But it is slower than the message passing updates. Also, according to Minka [2001], EP diverges for a good reason—indicating a poor approximating family or a poor energy function used by EP.

To address this issue, we propose a new approximate inference algorithm, Relaxed Expectation Propagation (REP). In REP, we introduce a relaxation factor  $r$  in the KL minimization used by EP (See Section 3) and penalize this relaxation factor. Because of this penalization, when the factor involved in the KL minimization is close to the current approximation, REP reduces to EP; when the factor is an outlier, the relaxation is used to stabilizing the message passing by relaxing the moment matching constraint. Regardless of the amount of outliers in data, REP converges in all of our experiments. To better understand REP, we also present the primal energy functions in Section 3. It differs from the EP energy function or the equivalent Bethe-like energy function [Heskes et al., 2005] by the use of relaxation factors.

To examine the performance of REP, in Section 5, we use it to train Gaussian process classification models for which EP is known to be a good choice for approximate inference [Kuss and Rasmussen, 2005]. In Section 7, we report experimental results on synthetic and UCI benchmark datasets, demonstrating that REP consistently outperforms EP and Power EP—in terms of algorithmic stability, estimation accuracy, and predictive performance.

## 2 Background: Expectation Propagation

Given observations  $\mathcal{D}$ , the posterior distribution of a probabilistic model with factors  $\{t_i(\mathbf{w})\}$  is

$$p(\mathbf{w}|\mathcal{D}) = \frac{1}{Z} \prod_{i=1, \dots, N} t_i(\mathbf{w}). \quad (1)$$

where  $Z$  is the normalization constant. Note that the prior distribution over  $\mathbf{w}$  is the factor  $t_0$  in the above equation and a factor  $t_i(\mathbf{w})$  may link to one, several, or all variables in  $\mathbf{w}$ . In general, we do not have a closed-form solution for the posterior calculation. We could use random sampling methods—such as the Metropolis Hasting—to obtain the posterior distribution, but these methods can suffer on slow convergence, especially for high dimensional problems.

To reduce the computational cost, [Minka \[2001\]](#) proposed EP to approximate the posterior distribution  $p(\mathbf{w}|D)$  by  $q(\mathbf{w})$  via factor approximation:

$$q(\mathbf{w}) = \prod_i \tilde{t}_i(\mathbf{w}) \quad (2)$$

where  $\tilde{t}_i(\mathbf{w})$  approximates  $t_i(\mathbf{w})$  and has a simpler tractable form. EP requires both  $q(\mathbf{w})$  and the approximation factor  $\tilde{t}_i(\mathbf{w})$  have the form of the exponential family—such as Gaussian or factorized (or some structured) discrete distributions. The approximation factors are unnormalized, but given them, we can also easily obtain the natural parameters of the approximate posterior  $q(\mathbf{w})$  due to the log linear property of the exponential family. For a graphical model representation, we can interpret the approximation factor  $\tilde{t}_i(\mathbf{w})$  as a message from the  $i^{th}$  exact factor  $t_i(\mathbf{w})$  to the variables linked to it.

To find the approximate posterior  $q$ , after initializing all the messages as one, EP iteratively refines the messages by repeating the following three steps: message deletion, belief projection, and message update, on each factor. In the message deletion step, we compute the partial posterior  $q^{\setminus i}(\mathbf{w})$  by removing a message  $\tilde{t}_i$  from the approximate posterior  $q^{\text{old}}(\mathbf{w})$ :  $q^{\setminus i}(\mathbf{w}) \propto q^{\text{old}}(\mathbf{w})/\tilde{t}_i(\mathbf{w})$ . In the projection step, we minimize the KL divergence between  $\hat{p}_i(\mathbf{w}) \propto t_i(\mathbf{w})q^{\setminus i}(\mathbf{w})$  and the new approximate posterior  $q(\mathbf{w})$ , such that the information from each factor is incorporated into  $q(\mathbf{w})$ . Finally, the message  $\tilde{t}_i$  is updated via  $\tilde{t}_i(\mathbf{w}) \propto q(\mathbf{w})/q^{\setminus i}(\mathbf{w})$ .

Since  $q(\mathbf{w})$  is in the exponential family, it has the following form

$$q(\mathbf{w}) \propto \exp(\boldsymbol{\nu}^T \boldsymbol{\phi}(\mathbf{w}))$$

where  $\boldsymbol{\phi}(\mathbf{w})$  are the features of the exponential family. Given this representation, the KL minimization in the key projection step is achieved by moment matching:

$$\int \boldsymbol{\phi}(\mathbf{w}) \hat{p}_i(\mathbf{w}) d\mathbf{w} = \int \boldsymbol{\phi}(\mathbf{w}) q(\mathbf{w}) d\mathbf{w} \quad (3)$$

This KL minimization distributed on each factor works very well, when the data is relatively clean and the approximate posterior  $q$  is not too far from  $\hat{p}_i$ . However, in practice, the presence of outliers can ruin the distributed KL minimization and leads to divergence of the algorithm.

### 3 Relaxed Expectation Propagation

In this section, we first present the new relaxed expectation propagation framework, discuss the choice of relaxation factors, and then describe its primal energy functions.

#### 3.1 The REP Algorithm

To reduce the impact of outlier factors, we can introduce relaxation factor  $r_i(w) \propto \exp(\boldsymbol{\eta}_i^T \phi(w))$  into the KL divergence. And to avoid too much relaxation we use a  $l_1$  penalty over it:

$$KL_r(\hat{p}_i r_i || q r_i) + c |\boldsymbol{\eta}_i|_1 \quad (4)$$

over  $q$  and  $r_i$ , where  $|\boldsymbol{\eta}_i|_1$  is the  $l_1$  norm of  $\boldsymbol{\eta}_i$ , the weight  $c$  controls how much relaxation we have, and the  $KL_r$  divergence is defined for unnormalized distributions.

This replacement allows us to adaptively handle factors—whether it is an outlier or not, and accurately approximate the posterior distribution  $p(\mathbf{w}|\mathcal{D})$  (1) by  $q(\mathbf{w}) \propto \prod_i \tilde{t}_i(\mathbf{w})$ .

With this relaxed KL divergence, we obtain the following REP algorithm:

1. Initialize  $q(\mathbf{w})$  as the prior  $t_0(\mathbf{w})$  (assuming the prior is in the exponential family) and all the messages  $\tilde{t}_i(\mathbf{w}) = 1$  for  $i = 1, \dots, N$ .
2. Loop until convergence or reaching the maximal number of iterations.

- Loop over factor  $i = 1, \dots, N$ :

- (a) **Message deletion:** Based on the current factor  $\tilde{t}_i$  and  $q^{\text{old}}$ , calculate the partial belief

$$q^{\setminus i} \propto q^{\text{old}}(\mathbf{w}) / \tilde{t}_i(\mathbf{w}).$$

- (b) **Belief projection:** Incorporate information from the exact factor  $t_i$  into the new belief  $q$  by minimizing the penalized KL:

$$\min_{r_i, q} KL_r(t_i r_i q^{\setminus i} || q r_i) + c |\boldsymbol{\eta}_i|_1 \quad (5)$$

where  $\hat{p}_i(\mathbf{w}) = t_i(\mathbf{w}) r_i(\mathbf{w}) q^{\setminus i}(\mathbf{w})$ .

- (c) **Message update:** Update the message based on the new belief:

$$\tilde{t}_i(\mathbf{w}) \propto q(\mathbf{w}) / q^{\setminus i}(\mathbf{w}).$$

Unlike EP, REP does not require strict moment matching between  $\hat{p}_i(\mathbf{w}) \propto t_i(\mathbf{w}) q^{\setminus i}(\mathbf{w})$  and the new approximate posterior  $q(\mathbf{w})$ . How close these moments are depends on how big  $\boldsymbol{\eta}_i$  is in the  $l_1$  penalized relaxation factor  $r_i$ .

#### 3.2 Choice of relaxation factors

For the relaxation factors  $r_i(\mathbf{w}) = \exp(\boldsymbol{\eta}_i^T \phi(\mathbf{w}))$ , we should parameterize  $\boldsymbol{\eta}_i$  in a form to make the minimization of (5) easy. Clearly, there are many choices available for us. A convenient one is to set (part of)  $\boldsymbol{\eta}_i$  to be a scaled version of the parameters of an old

message  $\tilde{t}_i$ , which can damp the influence of outliers via relaxed moment matching, but it will not cause double-counting of factors. The reason is that  $r_i$  appears in both sides of (5) and the new posterior  $q$  does not include  $r_i$ . With this choice, we can use moment matching to easily obtain an analytical solution for the product of  $q$  and  $r_i$ , greatly simplifying the joint optimization over  $q$  and  $r_i$ . This makes the computational overhead of REP over EP negligible in practice.

If we choose a form of  $r_i$  that makes the joint minimization over  $r_i$  and (i.e., belief)  $q$  expensive, we can still use a sequential minimization procedure: first minimize the penalized KL to obtain  $r_i$  based on the current  $q$ ; and then, based on the estimated relaxation factor, minimize the relaxed KL to obtain the new  $q$ .

### 3.3 Energy function

Now we give the primal and dual energy functions for relaxed expectation propagation. The primal energy function is

$$\begin{aligned} \min_{\boldsymbol{\eta}_i, \hat{p}_i} \max_q \sum_i \frac{1}{\hat{Z}_i} \int_{\mathbf{w}} \hat{p}_i(\mathbf{w}) r_i(\mathbf{w}) \log \frac{\hat{p}_i(\mathbf{w})}{\hat{Z}_i \tilde{t}_i(\mathbf{w}) p(\mathbf{w})} \\ - (n-1) \frac{1}{Z_q} \int_{\mathbf{w}} q(\mathbf{w}) r_i(\mathbf{w}) \log \frac{q(\mathbf{w})}{Z_q p(\mathbf{w})} + c \sum_i |\boldsymbol{\eta}_i| \end{aligned} \quad (6)$$

subject to

$$\frac{1}{\hat{Z}_i} \int_{\mathbf{w}} \phi(\mathbf{w}) \hat{p}_i(\mathbf{w}) r_i(\mathbf{w}) d\mathbf{w} = \frac{1}{Z_q} \int_{\mathbf{w}} \phi(\mathbf{w}) q(\mathbf{w}) r_i(\mathbf{w}) d\mathbf{w} \quad (7)$$

where  $\int_{\mathbf{w}} \hat{p}_i(\mathbf{w}) d\mathbf{w} = 1$ ,  $\int_{\mathbf{w}} q(\mathbf{w}) d\mathbf{w} = 1$ ,  $\hat{Z}_i = \int_{\mathbf{w}} \hat{p}_i(\mathbf{w}) r_i(\mathbf{w}) d\mathbf{w}$ , and  $Z_q = \int_{\mathbf{w}} q(\mathbf{w}) r_i(\mathbf{w}) d\mathbf{w}$ .

Based on the KL duality bound, we obtain the dual form of the energy function (See the Appendix for details). Setting the gradient of the dual function to zero gives us the fixed-point updates described in the previous section. The fixed-point updates, however, do not guarantee convergence, just like the classical EP updates. However, REP is much more robust than EP; in our experiments while EP diverges on difficult datasets, REP does *not* diverge in our experiments once.

We believe the robustness of REP comes from the relaxation of moment matching in (7): it does not demand the moments of  $\hat{p}_i$  and  $q$  to be exactly matched as in EP. Given an outlier factor, the exact moment matching requires the current  $q$  moves dramatically to a new  $q$ , ignoring all the information from the previous factors, summarized in the current  $q$ . And this can cause oscillations, reducing the final approximation accuracy.

From an optimization perspective, the min-max cost function (6) includes the cost function of EP as a special case by setting  $r_i(w) = 1$ . By tuning  $r_i(w)$ , it is possible to find a better solution to the min-max optimization. As shown by [Heskes et al. \[2005\]](#), the cost function of EP corresponds to the Bethe energy, an entropy approximation, with exact moment matching constraints. With relaxed moment matching, we can potentially obtain better entropy approximation (We will further our research along this line in the future).

Finally we want to stress that by REP robustifies EP to obtain an more accurate posterior approximation, rather than ignoring information from outliers, as shown in figure 1.

## 4 REP training for Gaussian process classification

In this section, we present a REP-based training algorithm for Gaussian process classification. First, let us denote  $N$  independent and identically distributed samples as  $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ , where  $\mathbf{x}_i$  is a  $d$  dimensional input and  $y_i$  is a scalar output. We assume there is a latent function  $f$  that we are modeling and the noisy realization of latent function  $f$  at  $\mathbf{x}_i$  is  $y_i$ .

We use a GP prior with zero mean over the latent function  $f$ . Its projection at the samples  $\{\mathbf{x}_i\}$  defines a joint Gaussian distribution:  $p(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, K)$  where  $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$  is the covariance function, which encodes the prior notation of smoothness. For classification, the data likelihood has the following form

$$p(y_i|f) = (1 - \epsilon)\Theta(f(\mathbf{x}_i)y_i) + \epsilon\Theta(-f(\mathbf{x}_i)y_i) \quad (8)$$

where  $\epsilon$  models the labeling error, and  $\Theta(a) = 1$  when  $a \geq 0$  ( $\Theta(a) = 0$  otherwise).

Given the GP prior over  $f$  and the data likelihood, the posterior process is

$$p(f|\mathcal{D}) \propto GP(f|0, K) \prod_{i=1}^N p(y_i|f(\mathbf{x}_i)) \quad (9)$$

Due to the nonlinearity in  $p(y_i|f)$ , the posterior process does not have a closed-form solution.

Using REP, we approximate each non Gaussian factor  $p(y_i|f(\mathbf{x}_i))$  by a Gaussian factor  $\tilde{t}_i(f_i) = \mathcal{N}(f_i|m_i, v_i)$ . Then we obtain a Gaussian process approximation to (9):

$$p(f|\mathcal{D}, \mathbf{t}) \propto GP(f|0, K) \prod_{i=1}^N \mathcal{N}(f_i|m_i, v_i) \quad (10)$$

We parameterize the relaxation factor  $r_i$  as an Gaussian:

$$r_i(f_i) \propto \mathcal{N}(f_i|m_i, b_i), \quad (11)$$

so that  $r_i$  share the mean as  $\tilde{t}_i$  and  $b_i$  is the only free parameter in  $r_i$ . For the convenience of the following presentation, we define  $\tilde{t}_{i,b}(f_i) \equiv \mathcal{N}(f_i|m_{i,b}, v_{i,b}) \propto r_i(f_i)\tilde{t}_i(f_i)$ . Now we give the relaxed EP algorithm for training a GP classifier.

1. Initialize  $m_i = 0$ ,  $v_i = \infty$ , and  $b_i = 0$  for  $\tilde{t}_i$ . Also, initialize  $r_i$ ,  $h_i = 0$ ,  $\mathbf{A} = \mathbf{K}$ , and  $\lambda_i = \mathbf{K}_{ii}$ .
2. Until all  $(m_i, v_i, b_i)$  converge: Loop  $i = 1, \dots, N$ :
  - (a) Remove  $\tilde{t}_i$  from the approximated posterior:

$$\lambda_i^{\setminus i} = \left(\frac{1}{\mathbf{A}_{ii}} - \frac{1}{v_i}\right)^{-1} \quad h_i^{\setminus i} = h_i + \lambda_i^{\setminus i} v_i^{-1} (h_i - m_i) \quad (12)$$

- (b) Minimize the relaxed KL divergence over  $b_i$  (i.e.,  $r_i$ ) by line search (See the Appendix).

(c) Multiple  $q^{\setminus i}$  with  $r_i$ :

$$\tilde{\lambda}_i^{\setminus i} = 1/(1/\lambda_i^{\setminus i} + b_i) \quad \tilde{h}_i^{\setminus i} = h_i^{\setminus i} - \tilde{\lambda}_i^{\setminus i} b_i (h_i^{\setminus i} - m_i) \quad (13)$$

(d) Minimize the relaxed KL divergence to obtain  $\tilde{t}_{i,b}$ :

$$\alpha = \frac{1}{\sqrt{\tilde{\lambda}_i^{\setminus i}}} \frac{(1 - 2\epsilon)\mathcal{N}(z|0, 1)}{\epsilon + (1 - 2\epsilon)\psi(z)} \quad \tilde{h}_i = \tilde{h}_i^{\setminus i} + \tilde{\lambda}_i^{\setminus i} \alpha \quad (14)$$

$$v_{i,b} = \tilde{\lambda}_i^{\setminus i} \left( \frac{1}{\alpha_i \tilde{h}_i} - 1 \right) \quad m_{i,b} = \tilde{h}_i + v_{i,b} \alpha \quad (15)$$

where  $z = \tilde{h}_i^{\setminus i} / \sqrt{\tilde{\lambda}_i^{\setminus i}}$  and  $\psi(\cdot)$  is the standard normal cumulative density distribution.

(e) Remove  $r_i$  from  $\tilde{t}_{i,b}$  to obtain  $\tilde{t}_i$ :

$$v_i = 1/(1/v_{i,b} + b_i) \quad m_i = v_i(m_{i,b}/v_{i,b} + m_i^{\text{old}} b_i) \quad (16)$$

(f) Update  $\mathbf{A}$  and  $h_i$ :

$$\mathbf{A} = \mathbf{A} - \frac{\mathbf{a}_i \mathbf{a}_i^T}{\delta + \mathbf{A}_{i,i}} \quad h_i = \sum_j \mathbf{A}_{ij} \frac{m_j}{v_j} \quad (17)$$

where  $\delta = 1/(1/v_i - 1/v_i^{\text{old}})$  and  $\mathbf{a}_i$  is the  $i$ -th column of  $\mathbf{A}$ .

We will release our software implementation upon the publication.

## 5 Related works

Minka [2005] proposed Power EP (PEP) via the use of the  $\alpha$ -divergence [Zhu and Rohwer, 1995]. The framework includes EP, fractional Belief propagation [Wiegerinck and Heskes, 2002], and variational Bayes as special cases, each of which is associated with a particular value  $\alpha$  in the  $\alpha$  divergence. In the presence of outliers, by using a power smaller than one for factors, Power EP increases the algorithmic stability over EP. But it also changes the divergence used for minimization to an  $\alpha$ -divergence that is different from KL, the desired divergence for many problems (e.g., classification). In contrast, REP adaptively relaxes the KL minimization for individual factors only when it becomes necessary.

We can damp the step size for message updates to help convergence, as suggested in [Minka, 2004]. But for difficult cases, we need to use a very small step size, greatly reducing the convergence speed. Furthermore, damping does not guarantee convergence. As a result, without using any stepsize, our approach is a good alternative to fix EP for difficult cases.



## 6 Experiments

In this section, we compare EP, PEP, and REP on approximation accuracy, convergence speed, and prediction accuracy for on Gaussian process classification. We chose GP classification as the test bed because EP has shown to be an excellent choice for approximation inference with GP classification models [Kuss and Rasmussen, 2005]. For EP, we used the updates described in Chapter 5.4 of the Thesis of Minka [2001]. Since there is no previous work that uses PEP for training GP, we derived the updates and described them in the Appendix. The reason we compared REP with PEP is because PEP can also help stabilize EP, possibly improving the approximation quality.

### 6.1 Evaluation of posterior approximation accuracy

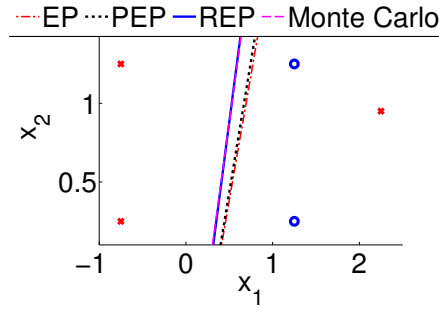
First, we considered linear classification of five data points shown in Figure 1. The red ‘x’ and blue ‘o’ data points belong two classes. The red point on the right is mislabeled. To reflect the true labeling error rate in the data, we set  $\epsilon = 0.2$  in (8). To obtain linear classifiers, we use the linear kernel— $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$  for the three algorithms. After the algorithms converge, we can obtain the posterior mean and covariance of a linear classifier  $\mathbf{w}$  in the 2-dimensional input space.

To measure the approximation quality, we first used importance sampling with  $10^8$  samples to obtain the exact posterior distribution of the classifier  $\mathbf{w}$ . We then applied these algorithms to obtain the approximate posteriors. We treated the (approximate) posterior means as the estimated classifiers and used them to generate their decision boundaries. They are visualized in Figure 1.a. For PEP, we set the power  $u$  to 0.8; for REP, we set  $c = 20$ . Given the outlier on the right, the EP decision boundary significantly differs from the exact Bayesian decision boundary obtained from the importance sampling; the PEP decision boundary is closer to the exact one; and the REP decision boundary overlaps with the exact one perfectly.

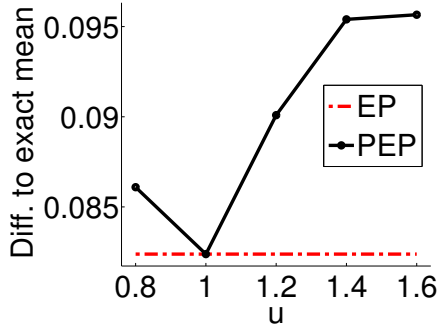
We also varied the relaxation weight  $c$  in (5) for REP and the power for PEP to examine their impact on approximate quality. We measured the mean square distances between the estimated and the exact mean vectors; we also computed the mean square distances between the estimated and the exact covariance matrices. The results are summarized in Figure 1.b to 1.e. For PEP, as shown in 1.b to 1.c, although the decision boundaries appear to be more aligned with the exact posterior distribution, their estimated mean and covariance are always worse than what EP achieve. This suggests that although PEP does reduce the influence of the outlier, it does not provide better approximation. By contrast, for REP, when  $c$  is big, the  $l_1$  penalty forces the relaxation factor  $b_i = 0$  (i.e.,  $r_i = 1$ ) and, accordingly, REP reduces to EP and gives the same results; And when  $c$  is relatively smaller (for a wide range of values), REP not only is immune to the presence of the outlier, but also improves the the approximation quality significantly.

Finally, we emphasize that REP aims to provide an accurate posterior approximation, regardless of likelihoods we used. For example, with various values of  $\epsilon$  (e.g., 0.1 and 0.25) in (8), REP consistently provides more accurately results than EP and PEP.

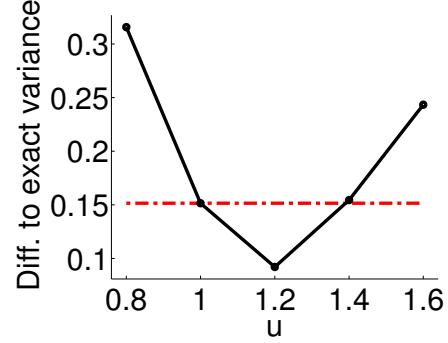




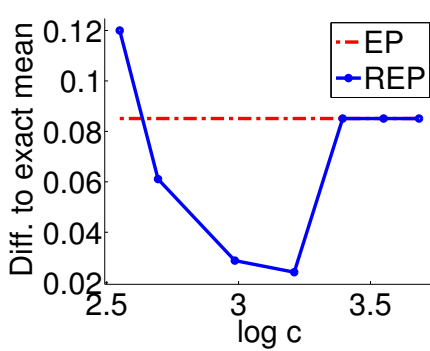
(a) Decision Boundaries



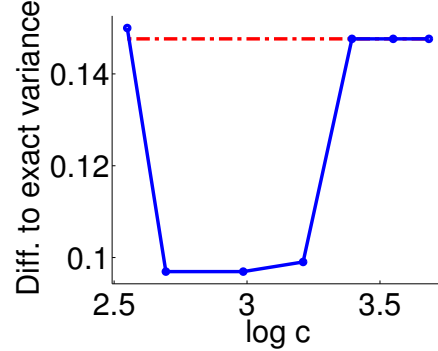
(b) Error in mean est.



(c) Error in var. est.



(d) Error in mean est.



(e) Error in var. est.

Figure 1: Classification of five data points, among which the red data point on the right is mislabeled. (a): Decision boundaries of EP, Power EP, and Relaxed EP; (b) and (c): EP vs Power EP with different powers  $u$ ; (d) and (e): EP vs Relaxed EP with different penalty weights  $c$ . REP reduces to EP when  $c$  is big. For a wide range of  $c$  values, the REP's approximation accuracy is significantly higher than those of EP and Power EP.

## 6.2 Results on synthetic data

We then compared these algorithms on a nonlinear classification task. We sampled 200 data points for each class: for class 1 the points were sampled from a single Gaussian distribution and, for class 2, the points from a mixture of two Gaussian components. The data points are represented by red crosses and blue circles for the two classes (See Figure 2). We randomly

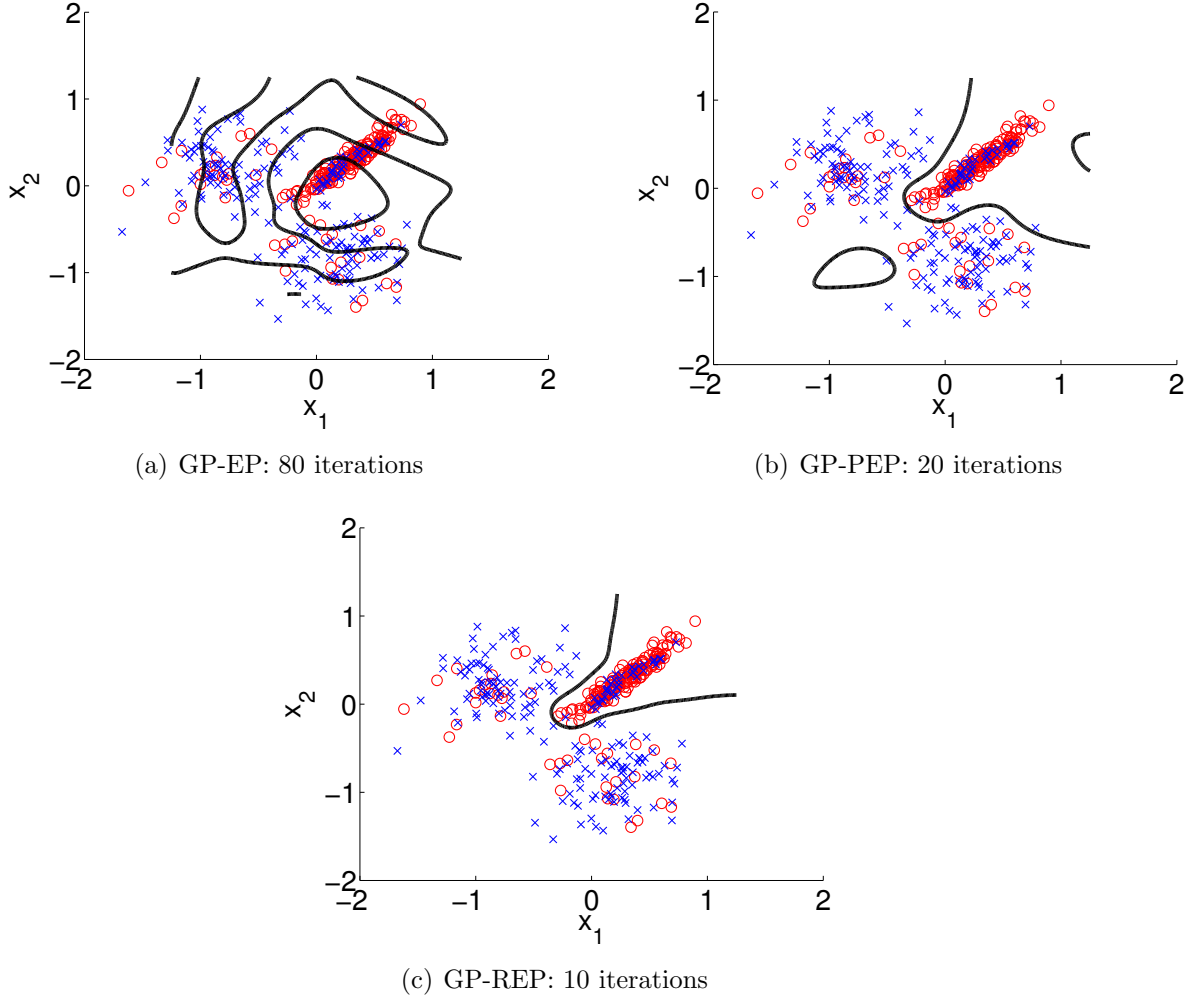


Figure 2: Decision boundaries of EP, Power EP, and REP. 20% of the data points are mislabeled.

flipped the labels of some data points to introduce labeling errors; we varied the error rates from 10% to 20%. And for each case, we let  $\epsilon$  match the error rate. We used a Gaussian kernel for all these training algorithms and applied cross-validation on the training data to tune the kernel width. We also tuned the relaxation weight  $c$  for REP and the power for PEP.

In Figure 2, we visualized the decision boundaries EP, PEP, and REP on one of the datasets with 20% labeling errors. To obtain these results, we set the power  $u = 0.8$  for Power EP and  $c = 10$  for Relaxed EP. Clearly, EP diverges and leads to a chaotic decision boundary. PEP converges in 20 iterations and gives a decision boundary—better than that of EP but with strange shapes. Finally, REP converges in only 10 iterations and provides a much more reasonable decision boundary than PEP.

To illustrate the convergence of PEP and REP, we visualized in Figure 3 the change of the GP parameter  $\alpha$  along iterations:  $R(iter) \equiv \|\alpha_{iter} - \alpha_{iter-1}\|_2$ . Clearly, PEP and REP are stabler than EP whose estimates oscillate—reflected by pikes in the  $R$  curve.

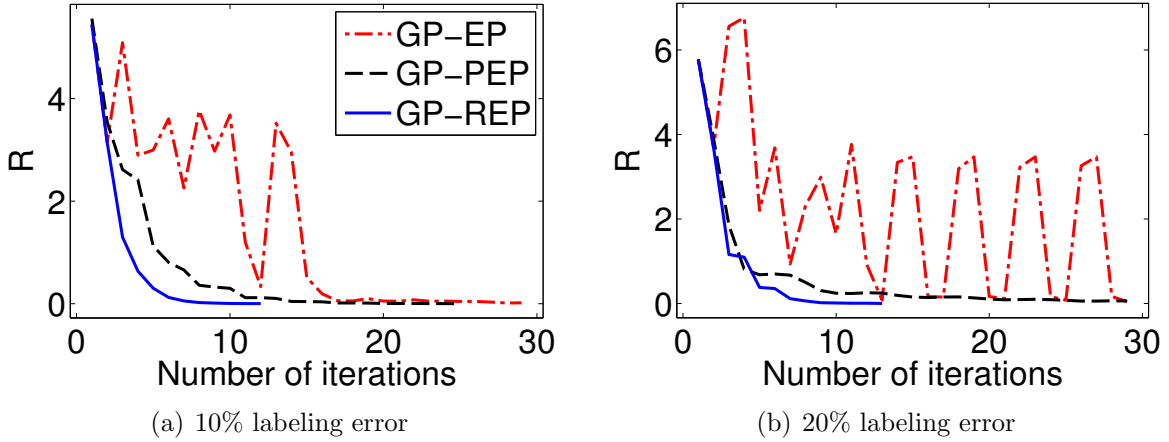


Figure 3: Change in GP parameters along iterations.

We repeated the experiments 10 times; each time we sampled 400 training and 39,600 test points. Figure 4 summarizes the results. Figure 4.a shows that the number of iterations before convergence. The results are averaged over 10 runs. To reach the convergence, we required  $R < 10^{-3}$ . Clearly, REP converges faster than PEP and EP.

Figure 4.b shows that while EP and PEP can diverge (PEP diverges less frequently than EP), REP *always* converges. Figure 4.c shows that REP gives significantly higher prediction accuracies than EP and PEP. Note that here we did not randomly flip the labels to introduce labeling errors in the test data and the prediction errors can be lower than the labeling errors in the training sets.

### 6.3 Results on real data

Finally we tested these algorithms on five UCI benchmark datasets: Heart, Pima, Diabetes, Haberman, and Spam.

For the Heart dataset, the task is to detect heart diseases with 13 features per sample. We randomly split the dataset into 81 training and 189 test samples 20 times. For the Pima dataset, we randomly split it into 319 training and 213 test samples, again 20 times. For the Diabetes dataset, medical measurements and personal history are used to predict whether a patient is diabetic. [Rätsch et al. \[2001\]](#) split the UCI Diabetes dataset into two groups (468 training and 300 test samples) for 100 times. We used the same partitions in our experiments. For the Haberman’s survival dataset, the task is to estimate whether the patient survive more than five years (including 5 years) after a surgery for breast cancer. The whole dataset contains information from 306 patient samples and 3 attributes per sample. We randomly split the dataset into 183 training and 123 test samples 100 times. Note that we did *not* add any labeling errors to these four datasets. Figure 5 summarizes the results. The prediction accuracies of GP-EP and GP-REP are averaged over the splits of each dataset. REP outperforms the competing algorithms significantly.

For the Spam dataset, the task is to detect spam emails. We partitioned the dataset to have 276 training and 4325 test samples, and flipped the labels of randomly selected data

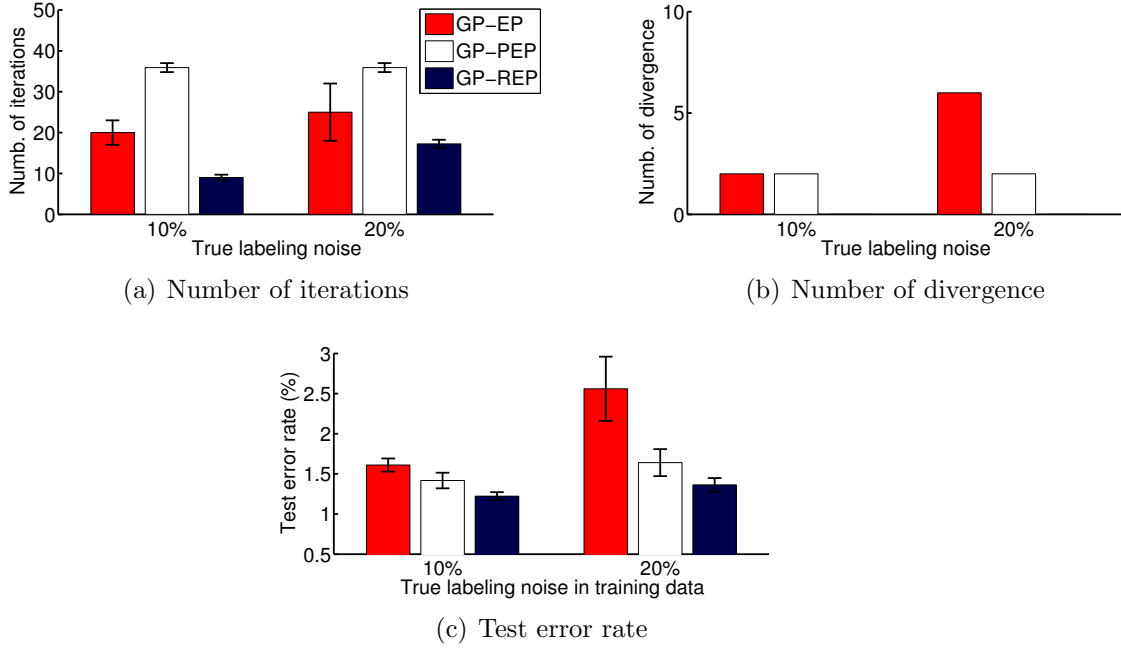


Figure 4: Comparison of EP, Power EP, and Relaxed EP on two datasets with different labeling noise levels. Relaxed EP always converges. And with fewer iterations, Relaxed EP consistently achieves higher prediction accuracies than EP and Power EP.

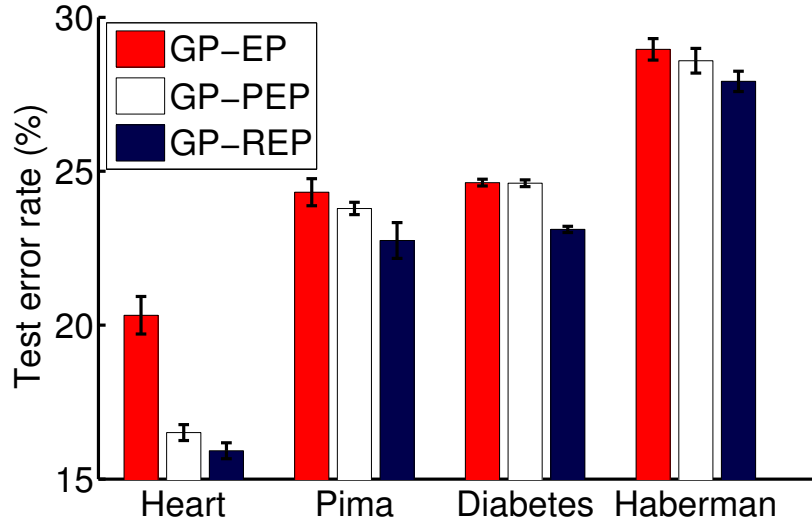


Figure 5: Test error rates of EP, PEP and REP on four UCI benchmark datasets without additional labeling noise.

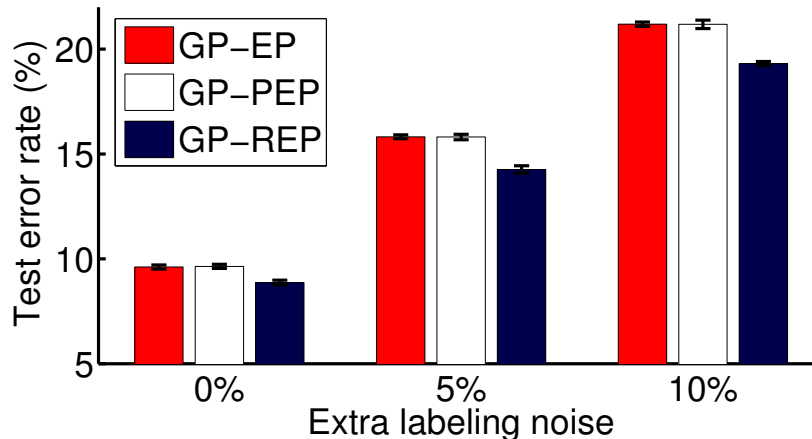


Figure 6: Test error rates of EP, PEP and REP on Spam dataset. We flipped the labels of some randomly selected data points to examine how these algorithms perform with outliers.

points from both the training and test samples. The experiment was repeated for 100 times. Figure 6 demonstrated that, with various additional labeling error rates, REP consistently achieves higher prediction accuracies than both EP and PEP.

## 7 Conclusions

In the paper we have introduced a method to increase the stability and approximation quality of EP. We relax the moment matching requirement of EP with a  $l_1$  penalty. Experimental results on GP classification demonstrate that the new inference algorithm avoids divergence and gives higher prediction accuracy than EP and Power EP.

## References

- Tommi S. Jaakkola. Tutorial on variational approximation methods. *In Advanced Mean Field Methods: Theory and Practice*, pages 129–159, 2000. doi: 10.1.1.31.8989.
- T.P. Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.
- Manfred Opper and Ole Winther. Expectation consistent approximate inference. *J. Mach. Learn. Res.*, 6:2177–2204, December 2005.
- Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, 1:1–305, January 2008.
- M. Kuss and C. Rasmussen. Assessing approximate inference for binary Gaussian process classification. *Journal of Machine Learning Research*, 6(10):1679–1704, 2005.

- A. L. Yuille. CCCP algorithms to minimize the bethe and kikuchi free energies: Convergent alternatives to belief propagation. *Neural Computation*, 14:2002, 2002.
- Tom Heskes, Manfred Oppel, Wim Wiegerinck, Ole Winther, and Onno Zoeter. Approximate inference techniques with expectation constraints. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(11):P11015, 2005.
- T. P. Minka. Divergence measures and message passing. Technical report, 2005.
- Huaiyu Zhu and Richard Rohwer. Bayesian invariant measurements of generalisation for continuous distributions. 1995.
- Wim Wiegerinck and Tom Heskes. Fractional belief propagation. In *in NIPS*, pages 438–445. MIT Press, 2002.
- T.P. Minka. Power EP. Technical Report MSR-TR-2004-149, Microsoft Research, Cambridge, January 2004.
- G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for adaboost. *Mach. Learn.*, 42: 287–320, March 2001.

# Appendices

## A Primal and dual energy functions for relaxed EP

The primary energy function of relaxed EP is

$$\begin{aligned} \min_{\boldsymbol{\eta}_i} \min_{\hat{p}_i} \max_q \sum_i \frac{1}{\hat{Z}_i} \int_{\mathbf{w}} \hat{p}_i(\mathbf{w}) r_i(\mathbf{w}) \log \frac{\hat{p}_i(\mathbf{w})}{\hat{Z}_i t_i(\mathbf{w}) p(\mathbf{w})} \\ - (n-1) \frac{1}{Z_q} \int_{\mathbf{w}} q(\mathbf{w}) r_i(\mathbf{w}) \log \frac{q(\mathbf{w})}{Z_q p(\mathbf{w})} + c \sum_i |\boldsymbol{\eta}_i|_1 \end{aligned} \quad (18)$$

subject to

$$\frac{1}{\hat{Z}_i} \int_{\mathbf{w}} \phi(\mathbf{w}) \hat{p}_i(\mathbf{w}) r_i(\mathbf{w}) d\mathbf{w} = \frac{1}{Z_q} \int_{\mathbf{w}} \phi(\mathbf{w}) q(\mathbf{w}) r_i(\mathbf{w}) d\mathbf{w} \quad (19)$$

$$\int_{\mathbf{w}} \hat{p}_i(\mathbf{w}) d\mathbf{w} = 1 \quad (20)$$

$$\int_{\mathbf{w}} q(\mathbf{w}) d\mathbf{w} = 1 \quad (21)$$

$$\hat{Z}_i = \int_{\mathbf{w}} \hat{p}_i(\mathbf{w}) r_i(\mathbf{w}) d\mathbf{w} \quad (22)$$

$$Z_q = \int_{\mathbf{w}} q(\mathbf{w}) r_i(\mathbf{w}) d\mathbf{w} \quad (23)$$

$$r_i(\mathbf{w}) \propto \exp(\boldsymbol{\eta}_i^T \phi(\mathbf{w})) \quad (24)$$

where  $c$  is the constant and  $r_i$  is the relaxation factor.

Based on the KL duality bound, we obtain the dual energy function.

$$\min_{\boldsymbol{\eta}} \min_{\boldsymbol{\nu}} \max_{\boldsymbol{\lambda}} (n-1) \log \int_{\mathbf{w}} p(\mathbf{w}) \exp(\boldsymbol{\nu}^T \phi(\mathbf{w}) + \boldsymbol{\eta}_i^T \phi(\mathbf{w})) d\mathbf{w} - \sum_{i=1}^n \log \int_{\mathbf{w}} t_i(\mathbf{w}) p(\mathbf{w}) \exp(\boldsymbol{\lambda}_i^T \phi(\mathbf{w}) + \boldsymbol{\eta}_i^T \phi(\mathbf{w})) d\mathbf{w} + c \sum_i |\boldsymbol{\eta}_i|_1 \quad (25)$$

$$(n-1)\boldsymbol{\nu} = \sum_i \boldsymbol{\lambda}_i \quad (26)$$

Setting the gradient of the above function to zero gives us the fixed-point updates described in the Section 3 of the main text. The fixed-point updates, however, do not guarantee convergence. But because of the relaxed KL minimization, REP always converges in our experiments (while EP can diverge when given many outliers).

Now we prove the duality of the relaxed EP energy function. Applying the KL duality to the first term in (6) produces

$$\begin{aligned} & \frac{1}{\hat{Z}_i} \int_{\mathbf{w}} \hat{p}_i(\mathbf{w}) r_i(\mathbf{w}) \log \frac{\hat{p}_i(\mathbf{w})}{\hat{Z}_i t_i(\mathbf{w}) p(\mathbf{w})} \\ &= \frac{1}{\hat{Z}_i} \int_{\mathbf{w}} \hat{p}_i(\mathbf{w}) r_i(\mathbf{w}) \log \frac{\hat{p}_i(\mathbf{w}) r_i(\mathbf{w})}{\hat{Z}_i t_i(\mathbf{w}) p(\mathbf{w}) r_i(\mathbf{w})} \\ &= \max_{\boldsymbol{\lambda}} \frac{1}{\hat{Z}_i} \int_{\mathbf{w}} \hat{p}_i(\mathbf{w}) r_i(\mathbf{w}) \boldsymbol{\lambda}_i(\mathbf{w}) d\mathbf{w} - \log \int_{\mathbf{w}} t_i(\mathbf{w}) p(\mathbf{w}) r_i(\mathbf{w}) \exp(\boldsymbol{\lambda}_i(\mathbf{w})) d\mathbf{w} \end{aligned} \quad (27)$$

This is because the maximum of the right side of (27) is achieved when (taking derivative to  $\boldsymbol{\lambda}_i(\mathbf{w})$ )

$$\frac{1}{\hat{Z}_i} \hat{p}_i(\mathbf{w}) r_i(\mathbf{w}) - \frac{t_i(\mathbf{w}) p(\mathbf{w}) r_i(\mathbf{w}) \exp(\boldsymbol{\lambda}_i(\mathbf{w}))}{\int_{\mathbf{w}} t_i(\mathbf{w}) p(\mathbf{w}) r_i(\mathbf{w}) \exp(\boldsymbol{\lambda}_i(\mathbf{w})) d\mathbf{w}} = 0 \quad (28)$$

which means

$$\exp(\boldsymbol{\lambda}_i(\mathbf{w})) = \frac{\hat{p}_i(\mathbf{w}) r_i(\mathbf{w}) \int_{\mathbf{w}} t_i(\mathbf{w}) p(\mathbf{w}) r_i(\mathbf{w}) \exp(\boldsymbol{\lambda}_i(\mathbf{w})) d\mathbf{w}}{\hat{Z}_i t_i(\mathbf{w}) p(\mathbf{w}) r_i(\mathbf{w})} \quad (29)$$

Inserting  $\exp(\boldsymbol{\lambda}_i(\mathbf{w}))$  in (27) proves the KL duality for (27).

And from the stationary condition, we can assume w.l.o.g. that

$$\boldsymbol{\lambda}_i(\mathbf{w}) = \boldsymbol{\lambda}_i^T \phi(\mathbf{w}) \quad (30)$$

$$\begin{aligned} & \frac{1}{\hat{Z}_i} \int_{\mathbf{w}} \hat{p}_i(\mathbf{w}) r_i(\mathbf{w}) \log \frac{\hat{p}_i(\mathbf{w})}{t_i(\mathbf{w}) p(\mathbf{w})} \\ &= \max_{\boldsymbol{\lambda}} \frac{1}{\hat{Z}_i} \int_{\mathbf{w}} \hat{p}_i(\mathbf{w}) r_i(\mathbf{w}) \boldsymbol{\lambda}_i^T \phi(\mathbf{w}) d\mathbf{w} - \log \int_{\mathbf{w}} t_i(\mathbf{w}) p(\mathbf{w}) r_i(\mathbf{w}) \exp(\boldsymbol{\lambda}_i^T \phi(\mathbf{w})) d\mathbf{w} \end{aligned} \quad (31)$$



Similarly, we have

$$\begin{aligned}
& -\frac{1}{Z_q} \int_{\mathbf{w}} q(\mathbf{w}) r_i(\mathbf{w}) \log \frac{q(\mathbf{w})}{Z_q p(\mathbf{w})} \\
&= -\frac{1}{Z_q} \int_{\mathbf{w}} q(\mathbf{w}) r_i(\mathbf{w}) \log \frac{q(\mathbf{w}) r_i(\mathbf{w})}{Z_q p(\mathbf{w}) r_i(\mathbf{w})} \\
&= \min_{\boldsymbol{\nu}} -\frac{1}{Z_q} \int_{\mathbf{w}} \boldsymbol{\nu}(\mathbf{w}) q(\mathbf{w}) r_i(\mathbf{w}) d\mathbf{w} + \log \int_{\mathbf{w}} p(\mathbf{w}) r_i(\mathbf{w}) \exp(\boldsymbol{\nu}(\mathbf{w})) d\mathbf{w} \\
&= \min_{\boldsymbol{\nu}} -\frac{1}{Z_q} \int_{\mathbf{w}} \boldsymbol{\nu}^T \phi(\mathbf{w}) q(\mathbf{w}) r_i(\mathbf{w}) d\mathbf{w} + \log \int_{\mathbf{w}} p(\mathbf{w}) r_i(\mathbf{w}) \exp(\boldsymbol{\nu}^T \phi(\mathbf{w})) d\mathbf{w}
\end{aligned} \tag{32}$$

With the constraint  $((n-1)\boldsymbol{\nu} = \sum_i \boldsymbol{\lambda}_i)$  and (2), we obtain the dual energy function:

$$\begin{aligned}
& \min_{\boldsymbol{\eta}} \min_{\boldsymbol{\nu}} \max_{\boldsymbol{\lambda}} (n-1) \log \int_{\mathbf{w}} p(\mathbf{w}) r_i(\mathbf{w}) \exp(\boldsymbol{\nu}^T \phi(\mathbf{w})) d\mathbf{w} \\
& - \sum_{i=1}^n \log \int_{\mathbf{w}} t_i(\mathbf{w}) p(\mathbf{w}) r_i(\mathbf{w}) \exp(\boldsymbol{\lambda}_i^T \phi(\mathbf{w})) d\mathbf{w} + c \sum_i |\boldsymbol{\eta}_i|_1
\end{aligned} \tag{33}$$

$$(n-1)\boldsymbol{\nu} = \sum_i \boldsymbol{\lambda}_i \tag{34}$$

## B Relaxed KL for GP classification

For GP classification, we minimize the relaxed KL divergence with  $l_1$  penalty over  $b_i$  by line search. Here we present how to compute the value of this cost function:

$$Q(b_i) = KL_r(t_i r_i q^{\setminus i} || r_i q) + c|b_i| \tag{35}$$

Following the notations in the main text (from equations (16) to (23)), we have  $Q(b_i)$  as

$$\begin{aligned}
& \frac{1}{\hat{Z}_i} \{[(1-\epsilon) \log(1-\epsilon) - \epsilon \log \epsilon] \psi(z) + \epsilon \log \epsilon\} + \frac{1}{2v_{i,b}} (F_{i,b} - \tilde{h}_i m_{i,b}) \\
& - \frac{1}{2} \log \left( 1 + (b_i + \frac{1}{v_{i,b}}) \lambda_i^{\setminus i} \right) + \frac{1}{2} \log(b_i \lambda_i^{\setminus i} + 1) - \frac{1}{2} b_i (m_i^2 - 2m_i \tilde{h}_i + F_{i,b}) \\
& + \frac{1}{2} \frac{(m_i - h_i^{\setminus i})^2}{\lambda_i^{\setminus i} + b_i^{-1}} - \log \hat{Z}_i + c|b_i|
\end{aligned} \tag{36}$$

where  $\hat{Z}_i = \epsilon + (1-2\epsilon)\psi(z)$ , and the term  $F_{i,b}$  can be computed as follows:

$$\delta_{i,b} = \left( \frac{1}{v_{i,b}} - \frac{1}{v_i} \right)^{-1} \tag{37}$$

$$a_{ii}^{new} = \left( \frac{1}{a_{ii}} + \frac{1}{\delta} \right)^{-1} \tag{38}$$

$$\tilde{a}_{ii}^{new} = a_{ii}^{new} \left( 1 - \frac{a_{ii}^{new}}{a_{ii}^{new} + b_i^{-1}} \right) \tag{39}$$

$$F_{i,b} = \tilde{a}_{ii}^{new} + \tilde{h}_i^2 \tag{40}$$

Using the above equations, we can efficiently optimize  $Q(b_i)$  over  $b_i$  via line search.

## C Power EP for GP classification

In this section, we describe how to train GP classifiers by Power EP. The updates of Power EP are the same as equations (5.64) to (5.74) in [Minka, 2001], except two critical modifications:

- Replace equation (5.67) in [Minka, 2001] by

$$\alpha_i = \frac{1}{\sqrt{\lambda_i}} \frac{[(1 - \epsilon)^u - \epsilon^u] \mathcal{N}(z|0, 1)}{\epsilon^u + [(1 - \epsilon)^u - \epsilon^u] \psi(z)} \quad (41)$$

where  $\psi(\cdot)$  is the standard normal cumulative density function and  $u$  is the power used by Power EP.

- Moreover, after (5.70), scale  $v_i$  by  $u$ :

$$v_i \leftarrow uv_i \quad (42)$$